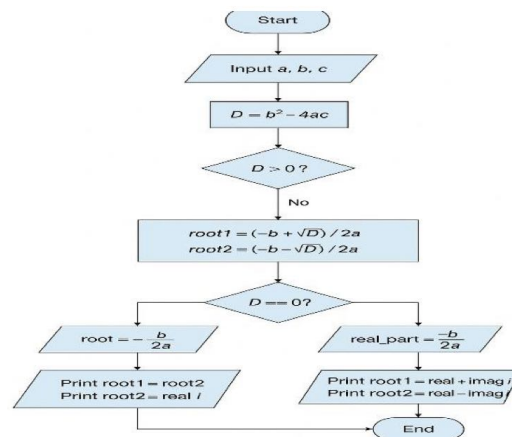


PROBLEM 2.1.1

Flowchart



Algorithm

Start

Input: Read three integers (a, b, and c) from a single line of input.

Calculate Discriminant: Compute D using the formula: $D = b^2 - 4ac$

- **If $D > 0$ (Real and Different):**
 - Calculate $\text{root1} = \frac{-b + \sqrt{D}}{2a}$
 - Calculate $\text{root2} = \frac{-b - \sqrt{D}}{2a}$
 - Print both roots.
- **If $D = 0$ (Real and Same):**
 - Calculate the single root: $\text{root} = \frac{-b}{2a}$
 - Print that $\text{root1} = \text{root2}$ equals this value.
- **If $D < 0$ (Imaginary/Complex):**
 - Calculate the **Real Part**: $\frac{-b}{2a}$
 - Calculate the **Imaginary Part**: $\frac{\sqrt{-D}}{2a}$
 - Print the roots in the complex format (e.g., real + imaginary i).

Formatting: Ensure all printed values are formatted to exactly two decimal places.

Stop

The screenshot shows a web-based IDE with a problem description on the left and a code editor on the right. The problem description for '2.1.1. Roots of a Quadratic Equation' includes the formula for the discriminant and instructions on how to handle different cases of D. The code in the editor implements this logic using Python, including input parsing, discriminant calculation, and conditional logic to print the roots in the correct format. The output shows the roots for a specific input, formatted to two decimal places.

```
1 import math
2 a, b, c = map(int, input().split())
3 D = b * b - 4 * a * c
4 if D > 0:
5     root1 = (-b + math.sqrt(D)) / (2 * a)
6     root2 = (-b - math.sqrt(D)) / (2 * a)
7     print(f"root1 = {root1:.2f}\nroot2 = {root2:.2f}")
8 elif D == 0:
9     root = -b / (2 * a)
10    print(f"root1 = root2 = {root:.2f}")
11 else:
12    real_part = -b / (2 * a)
13    imag_part = math.sqrt(-D) / (2 * a)
14    print(f"root1 = {real_part:.2f} + {imag_part:.2f}i\nroot2 = {real_part:.2f} - {imag_part:.2f}i")
```